

【特許請求の範囲】

【請求項 1】 プロセッサコアと、
可変な遅延値を持つ遅延回路を有するクロック信号線の
駆動回路と、
前記プロセッサコアにより前記遅延回路の遅延値を設定
するためのデータが書き込まれるレジスタと、
前記レジスタに書き込まれたデータに応じて前記遅延回
路の遅延値を制御する制御回路とを備えた半導体集積回
路。

【請求項 2】 プロセッサコアと、
可変な駆動能力を持つ出力バッファ回路を有するクロッ
ク信号線の駆動回路と、
前記プロセッサコアにより前記出力バッファ回路の駆動
能力を設定するためのデータが書き込まれるレジスタ
と、
前記レジスタに書き込まれたデータに応じて前記出力バ
ッファ回路の駆動能力を制御する制御回路とを備えた半
導体集積回路。

【請求項 3】 プロセッサコアが実行可能な命令コード
からなるソフトウェアを前記プロセッサコアに実行させ
ることにより、レジスタにデータを書き込むことを特徴
とする請求項 1 または請求項 2 記載の半導体集積回路。

【請求項 4】 所定回路のタイミングをテストするテス
ト用ソフトウェアと、このテスト用ソフトウェアを実行
したテスト結果に基づいて遅延回路の遅延値を制御す
るための駆動回路制御用ソフトウェアとを記憶するメモ
リを設け、
プロセッサコアが前記テスト用ソフトウェアおよび前記
駆動回路制御用ソフトウェアを実行するようしたことを
特徴とする請求項 1 記載の半導体集積回路。

【請求項 5】 所定回路のタイミングをテストするテス
ト用ソフトウェアと、このテスト用ソフトウェアを実行
したテスト結果に基づいて出力バッファ回路の駆動能力
を制御するための駆動回路制御用ソフトウェアとを記憶
するメモリを設け、
プロセッサコアが前記テスト用ソフトウェアおよび前記
駆動回路制御用ソフトウェアを実行するようしたことを
特徴とする請求項 2 記載の半導体集積回路。

【請求項 6】 可変な遅延値を持つ遅延回路を有するク
ロック信号線の駆動回路と、
複数の回路ブロックに含まれるフリップフロップにそれ
ぞれ入力されるクロック信号間のスキューを検出するク
ロックスキュー検出回路と、
前記クロックスキュー検出回路の検出する前記クロック
信号間のスキューに応じて前記遅延回路の遅延値を制御
する制御回路とを備えた半導体集積回路。

【請求項 7】 可変な駆動能力を持つ出力バッファ回路
を有するクロック信号線の駆動回路と、
複数の回路ブロックに含まれるフリップフロップにそれ
ぞれ入力されるクロック信号間のスキューを検出するク

ロックスキュー検出回路と、
前記クロックスキュー検出回路の検出する前記クロック
信号間のスキューに応じて前記出力バッファ回路の駆動
能力を制御する制御回路とを備えた半導体集積回路。

【請求項 8】 クロックスキュー検出回路は、
複数の回路ブロックのうち第 1 の回路ブロックに設けら
れ、前記第 1 の回路ブロックに含まれる他のフリップフ
ロップに入力されるクロック信号と同位相のクロック信
号を入力する第 1 のテスト用フリップフロップと、
前記複数の回路ブロックのうち前記第 1 の回路ブロック
の出力信号を入力する第 2 の回路ブロックに設けられ、
前記第 2 の回路ブロックに含まれる他のフリップフロ
ップに入力されるクロック信号と同位相のクロック信号を
入力するとともに前記第 1 のテスト用フリップフロップ
の出力データを入力データとする第 2 のテスト用フリ
ップフロップと、
前記第 1 のテスト用フリップフロップへデータを出力す
るデータ発生回路と、
前記第 1 のテスト用フリップフロップの出力データと前
記第 2 のテスト用フリップフロップの出力データとを用
いて前記第 1 のテスト用フリップフロップと前記第 2 の
テスト用フリップフロップとに入力されるクロック信号
間のスキューを検出するデータ解析回路とからなること
を特徴とする請求項 6 または請求項 7 記載の半導体集積
回路。

【請求項 9】 可変な遅延値を持つ遅延回路を有するク
ロック信号線の駆動回路と、
プロセス変動や使用時における電源電圧変動や温度変化
を反映した標準遅延値を検出する遅延評価回路と、
前記遅延評価回路の検出する前記標準遅延値に応じて前
記遅延回路の遅延値を制御する制御回路とを備えた半導
体集積回路。

【請求項 10】 可変な駆動能力を持つ出力バッファ回
路を有するクロック信号線の駆動回路と、
プロセス変動や使用時における電源電圧変動や温度変化
を反映した標準遅延値を検出する遅延評価回路と、
前記遅延評価回路の検出する前記標準遅延値に応じて前
記出力バッファ回路の駆動能力を制御する制御回路とを
備えた半導体集積回路。

【請求項 11】 遅延評価回路は、
遅延値が異なる複数の遅延回路と、
前記各遅延回路へデータを出力するデータ発生回路と、
前記各遅延回路の出力データを用いてプロセス変動や使
用時における電源電圧変動や温度変化を反映した標準遅
延値を検出するデータ解析回路とからなることを特徴と
する請求項 9 または請求項 10 記載の半導体集積回路。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、半導体集積回路に
関し、特に大規模システム L S I のクロック供給技術に

係わるものである。

【0002】

【従来の技術】システムLSI設計において現在主流となっている同期回路においては、チップ上の複数のフリップフロップに供給されるクロック信号のタイミングにわずかでもずれがあると、ただちに誤動作の原因になる。従って、クロックスキューを抑えることがLSI設計の大きな課題となっている。

【0003】従来の半導体集積回路では、クロックスキューの抑制は、例えば以下に示すような手順で行われることが多かった。まず、第一段階として、LSIを構成する各ブロックのレイアウト設計時に、当該ブロックのクロック入力端子からブロック内に含まれる各フリップフロップのクロック入力端子の間に、複数のバッファセルで構成されるクロックバッファツリーを作り込む。このときに、各バッファセルの駆動能力、各バッファセルが駆動する後段セルの入力容量、各バッファセルから後段セルに至る配線の負荷容量および抵抗値を考慮して、当該ブロックのクロック入力端子からブロック内に含まれる各フリップフロップのクロック入力端子までの遅延値を計算し、この遅延値がブロックに含まれる各フリップフロップ間で一定に揃うように、各バッファセルの駆動能力や各バッファセルと各フリップフロップの位置または配線の経路を調整する。これにより、各ブロック内ではフリップフロップ間でクロックスキューを抑制することが可能である。

【0004】次に、第二段階として、チップ全体のレイアウト設計時に、クロック生成回路から各ブロックのクロック入力端子に至る配線の負荷容量および抵抗値と各ブロック内でのクロック遅延値を考慮して、クロック生成回路から各ブロックのフリップフロップのクロック入力端子までの遅延値が一定に揃うように、クロック信号線駆動回路の駆動能力やクロック配線の負荷容量や抵抗値を調整する。これらはレイアウトデータを変更／カスタマイズすることにより実現する。これにより、最終的にチップ全体でフリップフロップ間のクロックスキューを抑制することができるのである。

【0005】

【発明が解決しようとする課題】以上述べたような従来の半導体集積回路には、以下に示すような課題がある。まず、半導体製造技術が進歩して回路が微細化されていくにつれて、回路遅延に対する配線の影響が大きくなってきており、従来に比べてタイミング設計が困難になってきている。

【0006】従来は、LSIを構成する各セルが駆動する負荷容量の中で、セルの入力容量が占める割合が相対的に大きかったため、配線の負荷容量は大きな影響を与えなかった。しかし、回路が微細化されるに伴い、配線の負荷容量の割合が増大しており、現在ではセルの入力容量と同程度になってきている。この傾向は今後も続く

ため、将来は配線が負荷容量の大きな部分を占めるようになる。

【0007】これは回路遅延を決めるもう一つの要因である抵抗値についても同様であり、今後は配線の負荷容量と抵抗値が回路遅延を決定することになる。このことがLSIのタイミング設計を困難にする理由は、今日のLSI設計では標準セルとしてあらかじめライブラリ化されたセルを用いるため、セルの入力容量は各チップ個別のレイアウト設計の影響を受けないのに対して、配線はレイアウト設計結果により全く変わってしまうためである。

【0008】そのために、現在ではレイアウト設計結果に基づいて回路遅延を計算するバックアノテーション技術を設計フローの中に取り込むのが一般的であるが、高精度が要求されるクロック信号に対してはバックアノテーション技術では精度が不十分である。そこでSPICEを用いたより高精度な解析が行われるが、それには大きな設計工数が必要となっている。

【0009】次に、最近の微細化技術の進歩に伴い、特定用途向けLSIは、従来のように汎用LSIの隙間を埋める小規模なグルーロジックからシステム全体を1チップに集積する大規模なシステムLSIへと変わってきている。このようなシステムLSIでは、設計工数の増大を避けるために、従来のように専用回路を最初から設計していくのではなく、可能な限り汎用回路をマクロとして再利用し、最小限必要な回路のみを専用設計するという設計手法を取る場合が多い。

【0010】このような状況では回路設計上の自由度が制限されるため、LSI設計全体としては設計工数を削減できるが、タイミング設計に関しては従来以上に難しくなる場合が多い。また現在では、回路規模の増大に伴い、LSIの消費電力の増大がLSI設計におけるもう一つの大きな問題となっている。この問題を解決するために、しばしば回路に供給するクロックを動的に制御し、動作する回路にのみクロックを供給して、動作しない回路にはクロックの供給を止めるというアプローチが取られることが多い。こうした場合、回路ブロックごとに異なるクロックを供給することが必要になり、前述のクロックスキューの抑制はさらに困難になる。

【0011】以上述べたようなクロック設計の難しさは、半導体製造技術の微細化に伴うプロセスばらつきの増大と合わさって、単に設計期間／工数の増大を招くだけでなく、設計ミスの増大につながり、試作したLSIの誤動作や動作マージン不足を引き起こす要因になる。本発明の目的は、以上述べたようなクロック設計上の課題を解決して、LSIの設計工数を削減できるとともに、クロック信号に係わるタイミングエラーを回避して誤動作を防止できる半導体集積回路を提供することである。

【0012】

【課題を解決するための手段】請求項1記載の半導体集積回路は、プロセッサコアと、可変な遅延値を持つ遅延回路を有するクロック信号線の駆動回路と、プロセッサコアにより遅延回路の遅延値を設定するためのデータが書き込まれるレジスタと、レジスタに書き込まれたデータに応じて遅延回路の遅延値を制御する制御回路とを備えている。

【0013】この構成によれば、プロセッサコアによりレジスタに書き込まれたデータに応じてクロック信号線の駆動回路中の遅延回路の遅延値を制御するようにしたことにより、クロック信号に係わるタイミングエラーを回避することができる。従って、従来のように回路修正／再試作を行う必要がなくなり、開発期間／工数を大幅に削減することが可能である。また、製造時の素子特性のばらつきやチップ使用時の電源電圧変動による遅延特性の変化を補償することができるので、チップ歩留まりを改善することも可能である。

【0014】請求項2記載の半導体集積回路は、プロセッサコアと、可変な駆動能力を持つ出力バッファ回路を有するクロック信号線の駆動回路と、プロセッサコアにより出力バッファ回路の駆動能力を設定するためのデータが書き込まれるレジスタと、レジスタに書き込まれたデータに応じて出力バッファ回路の駆動能力を制御する制御回路とを備えている。

【0015】この構成によれば、プロセッサコアによりレジスタに書き込まれたデータに応じてクロック信号線の駆動回路中の出力バッファ回路の駆動能力を制御するようにしたことにより、クロック信号に係わるタイミングエラーを回避することができる。従って、従来のように回路修正／再試作を行う必要がなくなり、開発期間／工数を大幅に削減することが可能である。また、製造時の素子特性のばらつきやチップ使用時の電源電圧変動による遅延特性の変化を補償することができるので、チップ歩留まりを改善することも可能である。

【0016】請求項3記載の半導体集積回路は、請求項1または請求項2記載の半導体集積回路において、プロセッサコアが実行可能な命令コードからなるソフトウェアをプロセッサコアに実行させることにより、レジスタにデータを書き込むことを特徴とする。この構成によれば、ソフトウェアによる高度なプログラムに基づいてクロック信号線の遅延を制御することにより、高精度なタイミング調整が可能になる。

【0017】請求項4記載の半導体集積回路は、請求項1記載の半導体集積回路において、所定回路のタイミングをテストするテスト用ソフトウェアと、このテスト用ソフトウェアを実行したテスト結果に基づいて遅延回路の遅延値を制御するための駆動回路制御用ソフトウェアとを記憶するメモリを設け、プロセッサコアがテスト用ソフトウェアおよび駆動回路制御用ソフトウェアを実行するようしたことを特徴とする。

【0018】この構成によれば、チップ上に搭載しているメモリに記憶したテスト用ソフトウェアをプロセッサコアが実行し、そのテスト結果に基づいてメモリ上に記憶した駆動回路制御用ソフトウェアを実行して遅延回路の遅延値の制御によりクロック信号線の遅延を制御することにより、自立的で回路動作中にも実行可能な実時間性を持ったタイミング調整を実現できる。

【0019】請求項5記載の半導体集積回路は、請求項2記載の半導体集積回路において、所定回路のタイミングをテストするテスト用ソフトウェアと、このテスト用ソフトウェアを実行したテスト結果に基づいて出力バッファ回路の駆動能力を制御するための駆動回路制御用ソフトウェアとを記憶するメモリを設け、プロセッサコアがテスト用ソフトウェアおよび駆動回路制御用ソフトウェアを実行するようしたことを特徴とする。

【0020】この構成によれば、チップ上に搭載しているメモリに記憶したテスト用ソフトウェアをプロセッサコアが実行し、そのテスト結果に基づいてメモリ上に記憶した駆動回路制御用ソフトウェアを実行して出力バッファ回路の駆動能力の制御によりクロック信号線の遅延を制御することにより、自立的で回路動作中にも実行可能な実時間性を持ったタイミング調整を実現できる。

【0021】請求項6記載の半導体集積回路は、可変な遅延値を持つ遅延回路を有するクロック信号線の駆動回路と、複数の回路ブロックに含まれるフリップフロップにそれぞれ入力されるクロック信号間のスキューを検出するクロックスキュー検出回路と、クロックスキュー検出回路の検出するクロック信号間のスキューに応じて遅延回路の遅延値を制御する制御回路とを備えている。

【0022】この構成によれば、クロックスキュー検出回路により、複数の回路ブロックに含まれるフリップフロップにそれぞれ入力されるクロック信号間のスキューを検出し、その検出したクロック信号間のスキューに応じてクロック信号線の駆動回路中の遅延回路の遅延値を制御するようにしたことにより、クロック信号に係わるタイミングエラーを回避することができる。従って、従来のように回路修正／再試作を行う必要がなくなり、開発期間／工数を大幅に削減することが可能である。また、製造時の素子特性のばらつきやチップ使用時の電源電圧変動による遅延特性の変化を補償することができるので、チップ歩留まりを改善することも可能である。

【0023】請求項7記載の半導体集積回路は、可変な駆動能力を持つ出力バッファ回路を有するクロック信号線の駆動回路と、複数の回路ブロックに含まれるフリップフロップにそれぞれ入力されるクロック信号間のスキューを検出するクロックスキュー検出回路と、クロックスキュー検出回路の検出するクロック信号間のスキューに応じて出力バッファ回路の駆動能力を制御する制御回路とを備えている。

【0024】この構成によれば、クロックスキュー検出

回路により、複数の回路ブロックに含まれるフリップフロップにそれぞれ入力されるクロック信号間のスキューを検出し、その検出したクロック信号間のスキューに応じてクロック信号線の駆動回路中の出力バッファ回路の駆動能力を制御するようにしたことにより、クロック信号に係わるタイミングエラーを回避することができる。従って、従来のように回路修正／再試作を行う必要がなくなり、開発期間／工数を大幅に削減することが可能である。また、製造時の素子特性のばらつきやチップ使用時の電源電圧変動による遅延特性の変化を補償することができるので、チップ歩留まりを改善することも可能である。

【0025】請求項8記載の半導体集積回路は、請求項6または請求項7記載の半導体集積回路において、クロックスキュー検出回路は、複数の回路ブロックのうち第1の回路ブロックに設けられ、第1の回路ブロックに含まれる他のフリップフロップに入力されるクロック信号と同位相のクロック信号を入力する第1のテスト用フリップフロップと、複数の回路ブロックのうち第1の回路ブロックの出力信号を入力する第2の回路ブロックに設けられ、第2の回路ブロックに含まれる他のフリップフロップに入力されるクロック信号と同位相のクロック信号を入力するとともに第1のテスト用フリップフロップの出力データを入力データとする第2のテスト用フリップフロップと、第1のテスト用フリップフロップへデータを出力するデータ発生回路と、第1のテスト用フリップフロップの出力データと第2のテスト用フリップフロップの出力データとを用いて第1のテスト用フリップフロップと第2のテスト用フリップフロップとに入力されるクロック信号間のスキューを検出するデータ解析回路とからなることを特徴とする。

【0026】このように、クロックスキュー検出回路を構成すればよい。請求項9記載の半導体集積回路は、可変な遅延値を持つ遅延回路を有するクロック信号線の駆動回路と、プロセス変動や使用時における電源電圧変動や温度変化を反映した標準遅延値を検出する遅延評価回路と、遅延評価回路の検出する標準遅延値に応じて遅延回路の遅延値を制御する制御回路とを備えている。

【0027】この構成によれば、遅延評価回路により、プロセス変動や使用時における電源電圧変動や温度変化を反映した標準遅延値を検出し、その検出した標準遅延値に応じてクロック信号線の駆動回路中の遅延回路の遅延値を制御するようにしたことにより、クロック信号に係わるタイミングエラーを回避することができる。従って、従来のように回路修正／再試作を行う必要がなくなり、開発期間／工数を大幅に削減することが可能である。また、製造時の素子特性のばらつきやチップ使用時の電源電圧変動による遅延特性の変化を補償することができるので、チップ歩留まりを改善することも可能である。

【0028】請求項10記載の半導体集積回路は、可変な駆動能力を持つ出力バッファ回路を有するクロック信号線の駆動回路と、プロセス変動や使用時における電源電圧変動や温度変化を反映した標準遅延値を検出する遅延評価回路と、遅延評価回路の検出する標準遅延値に応じて出力バッファ回路の駆動能力を制御する制御回路とを備えている。

【0029】この構成によれば、遅延評価回路により、プロセス変動や使用時における電源電圧変動や温度変化を反映した標準遅延値を検出し、その検出した標準遅延値に応じてクロック信号線の駆動回路中の出力バッファ回路の駆動能力を制御するようにしたことにより、クロック信号に係わるタイミングエラーを回避することができる。従って、従来のように回路修正／再試作を行う必要がなくなり、開発期間／工数を大幅に削減することが可能である。また、製造時の素子特性のばらつきやチップ使用時の電源電圧変動による遅延特性の変化を補償することができるので、チップ歩留まりを改善することも可能である。

【0030】請求項11記載の半導体集積回路は、請求項9または請求項10記載の半導体集積回路において、遅延評価回路は、遅延値が異なる複数の遅延回路と、各遅延回路へデータを出力するデータ発生回路と、各遅延回路の出力データを用いてプロセス変動や使用時における電源電圧変動や温度変化を反映した標準遅延値を検出するデータ解析回路とからなることを特徴とする。

【0031】このように、遅延評価回路を構成すればよい。

【0032】

【発明の実施の形態】以下、本発明の実施の形態について図面に基づいて説明する。

〔第1の実施の形態〕図1は本発明の第1の実施の形態の半導体集積回路の構成図である。図中、1はクロック原信号生成回路であり、LSIに接続される水晶発振器が出力する発振信号を整形し、必要に応じて周波数を過倍するなどしてクロックの原信号を生成する。2は遅延手段である。この遅延手段2は、図に示すように、インバータ回路を複数段接続した構成であり、インバータ回路2段毎に信号を出力しており、この各々の出力が、入力されるクロック原信号を異なった時間だけ遅延させた信号になっている。もちろん、ここで示した遅延手段2以外にも様々な遅延手段の例を考えることができ、本発明はそれらも含んでいることは言うまでもない。3は選択回路であり、遅延手段2の複数の出力信号の中から1つを選択して出力することにより、クロック信号線5の遅延値を変更することができるようになっている。2Aは前述の遅延手段2および選択回路3からなる遅延回路であり、入力されるクロック原信号を遅延させて出力し、その遅延時間を変更できるものである。

【0033】4は出力バッファ回路である。出力バッファ

ァ回路 4 の出力はチップ上の特定ブロックにクロックを供給するクロック信号線 5 であるが、通常、クロック信号線 5 はブロック間を長い距離配線されるので、負荷容量および抵抗値が大きくなってしまふ。そのため、出力バッファ回路 4 には負荷に相当する大きさの駆動能力を持たせないとクロック信号が訛ってしまい、その結果、クロックスキューが大きくなってしまふ。また、出力バッファ回路 4 の駆動能力を変えることにより、クロック信号線 5 の遅延値を調整することができる。本実施の形態における出力バッファ回路 4 は、複数の 3 ステートバッファ 4 a、4 b、4 c を並列に接続した構成になっており、同時にオンさせる 3 ステートバッファの個数を変更することで、クロック信号線 5 に対する駆動能力を変更することができるようになっている。なお、出力バッファ回路 4 内の 3 ステートバッファの個数は 3 個に限定されるものではなく、クロック信号線 5 の遅延値を調整するのに十分な複数個であればよい。

【0034】6 は CPU である。CPU 6 はシステム LSI に搭載されるプロセッサコアとしては最も一般的なものであり、通常、汎用の命令セットを持ち、この命令セットに含まれる命令を用いて作成されたソフトウェアを順次実行することにより、様々な応用に対応した信号処理を実現することができる。7 はメモリである。メモリ 7 は比較的大規模な記憶装置であり、前記ソフトウェアや信号処理を行う上での中間データなどを記憶させるのに用いる。本実施の形態では、メモリ 7 に格納されるソフトウェアとして、タイミングテストプログラム 11 と駆動回路制御プログラム 12 を含んでいることが特徴である。

【0035】8 はレジスタである。レジスタ 8 も記憶装置の一種であるが、メモリ 7 と比べると記憶容量が小さく、特定のデータを記憶するのに用いられる。本実施の形態では、選択回路 3 の機能や出力バッファ回路 4 の能力を選択するためのデータを記憶させるのに用いている。9 はランダムロジック（制御回路）であり、レジスタ 8 に格納されたデータをデコードして選択回路 3 や出力バッファ回路 4 の制御信号を出力する。

【0036】13 および 15 もレジスタである。レジスタ 13 にはクロック 14 が入力される。また、レジスタ 15 にはクロック 16 が入力される。各レジスタ 13、15 はそれぞれに入力されるクロック 14、16 の立ち上がりエッジに同期してデータを取り込む。17 は被テスト回路である。現在主流となっている同期設計では、組み合わせ回路からなる被テスト回路 17 はレジスタ 13 からのデータを受けて、何らかの処理を加えた後、レジスタ 15 にデータを書き込むことを行う。本実施の形態で、被テスト回路 17 になるのは、チップ上でタイミング的に厳しい部分である。これらの被テスト回路 17 は回路設計段階で見つけ出すことができる。なお、被テスト回路 17 は、半導体集積回路の主要回路のうちの

部として存在するが、レジスタ 13、15 は半導体集積回路の主要回路のうちの一部として存在するものを用いる場合もあるし、テスト専用として設ける場合もある。

【0037】CPU 6 とメモリ 7 やレジスタ 8、レジスタ 13、レジスタ 15 はバス信号線 10 により相互に接続されている。本実施の形態では直接関係しないので記述していないが、通常はそれ以外にも複数の周辺回路がバス信号線 10 に接続されることが多い。また、メモリ 7 と同じくレジスタ 8、レジスタ 13、レジスタ 15 にもアドレスが割り振られているので、CPU 6 はメモリ 7 にデータを記憶させるのと全く同じようにして、ソフトウェアを実行する中でレジスタ 8、レジスタ 13、レジスタ 15 にデータを記憶させることができる。

【0038】なお、クロック原信号生成回路 1 はチップ上に複数設けてあってもよいが、ここでは、1 個設けてあるものとする。クロック原信号生成回路 1 からは複数のクロック原信号が出力され、各々のクロック原信号はそれぞれの駆動回路を通して異なるクロック信号として各回路ブロックへ供給される。図 1 では、1 つのクロック信号（クロック信号線 5 のクロック信号）の駆動回路として、遅延回路 2 A と出力バッファ回路 4 とを有し、それらを制御するためにレジスタ 8 およびランダムロジック 9 を設けている。なお、他のクロック信号については、上記のクロック信号線 5 のクロック信号と同様に遅延調整機能を持つ駆動回路（遅延回路 2 A、出力バッファ回路 4）やレジスタ 8 およびランダムロジック 9 を設けてあってもよいし、あるクロック信号については、遅延調整機能を持たない構成の駆動回路としてもよい。

【0039】また、本実施の形態において、クロック 16 はクロック信号線 5 から供給されるクロック信号であり、クロック 14 は図示しない他のクロック信号線から供給されるクロック信号であるとする。上記のような構成を持つ本実施の形態の半導体集積回路におけるクロック調整は、図 2 に示す設計フローにより実現できる。

【0040】本実施の形態の半導体集積回路（LSI）のレイアウト設計において、各ブロックのレイアウトとクロック信号線 5 を含むブロック間配線のレイアウトが決まった段階（ステップ 61）で、ブロック内部のクロック遅延値（各ブロックのクロック入力端子からフリップフロップのクロック端子に至る経路の遅延値）とクロック信号線 5 の負荷容量および抵抗値とをレイアウトデータから見積る（ステップ 62）。これらの見積り値に基づいて、クロック原信号生成回路 1 の出力から各ブロックのフリップフロップのクロック端子に至る経路の遅延値が全てのブロック間で一定値に揃うように、クロック原信号生成回路 1 の出力からクロック信号線 5 までの目標遅延値を逆算して求める（ステップ 63）。

【0041】次にステップ 64 で、SPICE シミュレーション等の方法を用いて、クロック信号線 5 の遅延値が上記で算出した適切な値（目標遅延値）になるよう

に、遅延回路2Aの遅延値および出力バッファ回路4の駆動能力を合わせ込む。ここまでは本実施の形態でも従来方法でも同じである。しかし、本実施の形態では、この段階で従来のように、チップのレイアウトを修正してクロックの遅延値を調整することは行わない。その代わり、それを実現するようにレジスタ8の初期値を決める。従って、従来のレイアウト修正に要していた設計時間/工数を削減することができる。

【0042】その後、LSIを製造して(ステップ65)、テスターで評価するに際して、LSIのリセット後ただちに、CPU6がLSI初期化处理の一環として、あらかじめ求めていた初期値をレジスタ8に設定する。次に、CPU6がメモリ7に蓄積されているタイミングテストプログラム11を実行してチップの評価を行う(ステップ66)。ここで問題がなければ、チップの正常動作が確認できるはずである。しかしながら、もしもチップの設計時に行った遅延値の見積りが誤っていたり、精度的に不十分であった場合には、テストの結果、タイミングエラーのために、チップが誤動作するということが判明する場合もある。

【0043】もしもそうなった場合には、従来の設計方法だと、クロック信号線駆動回路のレイアウトを再修正して、クロックスキューを合わせ込むことが必要になり、LSI開発スケジュールを大幅に遅らせる結果になる。しかしながら、本実施の形態では、ステップ67で、CPU6がメモリ7内の駆動回路制御プログラム12を実行して、レジスタ8に設定する値(遅延回路2Aの遅延値、出力バッファ回路4の駆動能力を設定するためのデータ)を変更するだけで、チップの誤動作を解決することができ、ここでもLSI開発期間/工数の大幅な短縮を実現している。

【0044】次に、ステップ66のタイミングテストおよびステップ67の駆動回路制御について、図3のフロー図を用いてより詳しく説明する。タイミングテスト66は、あるクロックサイクルでCPU6が被テスト回路17の入力レジスタ13にテストデータを書きこむことから始まる(ステップ71)。次のクロックの立ち上がりにより同期してレジスタ13は入力データを取り込む。被テスト回路17はこの入力データと回路の論理によって決まる出力データのある遅延時間後に出力レジスタ15に対して出力する。

【0045】次のクロックの立ち上がりにより同期して被テスト回路17の出力データはレジスタ15に取り込まれる。この時、もしもタイミングエラーがなければ、レジスタ15には正しい出力データが取り込まれるが、もしもクロック14とクロック16間にスキューがあつてクロック16がクロック14に対して遅れているような場合には、レジスタ15には誤ったデータが格納されることになる。

【0046】次のクロックサイクルでCPU6はレジス

タ15から出力データを読み出し、一旦CPU6内部の汎用レジスタに格納しておく(ステップ72)。次にCPU6は、あらかじめ計算してタイミングテストプログラム11の中に含まれている正しいデータをメモリ7から読み出して、前記出力データと比較する(ステップ73)。比較の結果が一致していればタイミングエラーはないということでプログラムは終了する。

【0047】比較の結果が一致していない場合には、駆動回路制御67(図2)を行う。まず、出力データの誤りを分析する(ステップ74)。多くの回路ではスキュー時間と出力データの間に相関があるので、誤り方を分析することにより、スキュー時間を逆算することができる。次に、駆動回路制御プログラム12に含まれるデータを用いて、スキュー時間を補正するようにクロック16を早める駆動回路の新しい設定値を逆算する(ステップ75)。最後に、CPU6はクロック16を供給するクロック信号線5の駆動回路に対応するレジスタ8に新しい設定値を書き込む(ステップ76)。その結果、クロック16のタイミングが速くなるため、クロック14との間のスキューはなくなり、タイミングエラーは解消される。

【0048】以上のように本実施の形態によれば、CPU6によりレジスタ8に書き込まれたデータに応じてクロック信号線5の駆動回路を制御する、すなわち遅延回路2Aの遅延値および出力バッファ回路4の駆動能力を制御するようにしたことにより、クロック信号が係わるタイミングエラーを回避することができる。従って、従来のように回路修正/再試作を行う必要がなくなり、開発期間/工数を大幅に削減することが可能である。また、製造時の素子特性のばらつきやチップ使用時の電源電圧変動による遅延特性の変化を補償することができるので、チップ歩留まりを改善することも可能である。

【0049】また、本実施の形態では、遅延調整機能を持つ遅延回路2Aおよび出力バッファ回路4を設け、その両方を制御するようにしたが、遅延回路2Aおよび出力バッファ回路4のうち一方のみを設けて制御するようにしてもよく、その場合、構成および制御が簡単になる。また、本実施の形態では、例えば出力バッファ回路4の調整を主とし、それで調整できないときのみ遅延回路2Aにより調整するようにしてもよい。なお、遅延回路2Aおよび出力バッファ回路4のうち一方のみを設ける場合に比べ、本実施の形態のように両方設けた方が、遅延調整能力は増加するが制御および構成が複雑になる。

【0050】なお、本実施の形態では、タイミングエラーの例としてクロックスキューの問題を取り上げたが、それ以外にもクロックに係わる様々なタイミングの問題があり、それらに対しても本発明は有効である。例えば、被テスト回路17の遅延時間がクロックサイクル時間よりも大きい場合には、やはりレジスタ15は正しい

データを取り込めないのでタイミングエラーが起こることになる。このような場合には、逆にクロック 16 を遅らせて、わざとクロック 14 との間にスキューをつくることによりタイミングエラーを解消することが可能である。

【0051】なお、遅延回路 2 A の遅延値の制御によりクロック信号線 5 のクロック 16 のタイミングを速める場合には、選択回路 3 が、遅延手段 2 からの複数の入力のうち、クロック原信号生成回路 1 側に近い入力を選択するように、選択回路 3 を制御すればよいし、クロック信号線 5 のクロック 16 のタイミングを遅らせる場合には、選択回路 3 が、遅延手段 2 からの複数の入力のうち、クロック原信号生成回路 1 側から遠い入力を選択するように、選択回路 3 を制御すればよい。

【0052】また、出力バッファ回路 4 の駆動能力の制御によりクロック信号線 5 のクロック 16 のタイミングを速める場合には、3 ステートバッファ 4 a、4 b、4 c の制御入力をイネーブルに設定するものの数を増やせばよい。これは、オンする 3 ステートバッファの数が増えたと出力バッファ回路 4 の駆動能力が大きくなり、クロック信号を高速に駆動できるようになるためである。逆に、クロック信号線 5 のクロック 16 のタイミングを遅らせる場合には、3 ステートバッファ 4 a、4 b、4 c の制御入力をイネーブルに設定するものの数を減らせばよい。

【0053】また、本実施の形態では、被テスト回路 17 の入力レジスタ 13 と出力レジスタ 15 がともにバス信号線 10 に接続され、CPU 6 から直接読み書きできるような構成になっているが、必ずしもそうである必要はない。ただし、その場合にはテストデータをレジスタ 13 に書き込み、レジスタ 15 から結果データを読み出すのに、より複雑なシーケンスが必要になる。

【0054】また、レジスタ 13 やレジスタ 15 はより大規模なメモリであってもよい。

【第 2 の実施の形態】次に、本発明の第 2 の実施の形態について以下に説明する。図 4 は本発明の第 2 の実施の形態の半導体集積回路の構成図である。図 4 において、クロック原信号生成回路 1、遅延手段 2、選択回路 3、出力バッファ回路 4、クロック信号線 5 については、第 1 の実施の形態と基本的に同じであるため、説明を省略する。

【0055】本実施の形態においては、第 1 の実施の形態とは異なり、クロック信号線 5 の駆動回路（遅延回路 2 A、出力バッファ回路 4）は、ランダムロジック 44 を介してクロックスキュー検出回路 53 が出力するデータにより制御される。以下、クロックスキュー検出回路 53 について詳しく説明する。クロックスキュー検出回路 53 は、データ発生回路 45、テスト用のフリップフロップ 46、テスト用のフリップフロップ 49 およびデータ解析回路 52 等から構成される。データ発生回路 4

5 の出力はフリップフロップ 46 に入力され、フリップフロップ 46 の出力はフリップフロップ 49 およびデータ解析回路 52 に入力されている。また、フリップフロップ 49 の出力はデータ解析回路 52 に入力されている。

【0056】テスト用のフリップフロップ 46 は回路ブロック 48 に含まれてレイアウトされ、テスト用のフリップフロップ 49 は回路ブロック 51 に含まれてレイアウトされている。なお、回路ブロック 48、51 のそれぞれには、主要回路を構成する図示しないフリップフロップ（以下「他のフリップフロップ」という）が複数個レイアウトされており、回路ブロック 48 中のフリップフロップ 46 および他のフリップフロップには同一（同位相）のクロック 47 が入力され、回路ブロック 51 中のフリップフロップ 49 および他のフリップフロップには同一（同位相）のクロック 50 が入力される。回路ブロック 48、51 のそれぞれのブロック内に含まれる他のフリップフロップ間ではクロックスキューが十分小さくなるようにクロックバッファの構成やサイズが調整されている。

【0057】図 5 にクロックスキュー検出回路 53 の構成例を示す。データ発生回路 45 はクロックサイクル毎に変化する 1 ビットの信号を生成する回路であり、例えば図 5 に示すように、フリップフロップ 46 の出力を反転させるインバータ素子 81 を用いて構成することができる。この例では、データ発生回路 45 であるインバータ素子 81 が回路ブロック 48 に含まれてレイアウトされている。フリップフロップ 46 にはクロック 47 に同期して、各クロックサイクル毎に前サイクルとは異なる値がラッチされる。

【0058】フリップフロップ 46 の出力はフリップフロップ 49 に入力される。この配線はブロック 48、51 間をほぼ最短距離で結ぶようにレイアウトし、フリップフロップ 46、49 間の信号遅延が小さくなるようにする。このバス遅延値が、回路ブロック 48 に含まれる他の全フリップフロップから回路ブロック 51 に含まれる他の全フリップフロップに至る全ての信号バスの遅延値よりも小さくなるように、フリップフロップ 46 の駆動能力を選択する。

【0059】フリップフロップ 49 はクロック 50 に同期してフリップフロップ 46 の出力をラッチする。この時、クロック 47 とクロック 50 間のスキューが十分小さい場合には、フリップフロップ 49 には前サイクルにフリップフロップ 46 にラッチされた値が正しくラッチされることになる。従って、同一サイクルではフリップフロップ 46 とフリップフロップ 49 に異なる値が保持されることになる。

【0060】しかし、クロック 50 の立ち上がりがクロック 47 の立ち上がりに対してある遅延値以上遅れると、クロック 47 の立ち上がりに同期してフリップフロ

ップ 46 にラッチされた値が伝搬してフリップフロップ 49 まで到達した後でクロック 50 が立ち上がり、そのままフリップフロップ 49 にラッチされることになる。このように、タイミングエラーが起こると、同一サイクルでフリップフロップ 46 とフリップフロップ 49 に同じ値が保持されることになる。

【0061】また、データ解析回路 52 は、フリップフロップ 46 とフリップフロップ 49 に保持される値を比較する回路であり、例えば図 5 に示すように、フリップフロップ 46 の出力とフリップフロップ 49 の出力とを入力する EXNOR (排他的論理和の否定) 素子 82 を用いて構成することができる。この例では、データ解析回路 52 である EXNOR 素子 82 が回路ブロック 51 に含まれてレイアウトされている。EXNOR 素子 82 は、フリップフロップ 46 の出力とフリップフロップ 49 の出力とが一致していれば値 1 を、異なっていれば値 0 をクロックスキュー検出信号 83 として出力する。このクロックスキュー検出信号 83 の値が 1 のときにはクロックスキューによるタイミングエラーが起こっており、0 のときにはタイミングエラーは起こっていない。

【0062】ブロック 48 およびブロック 51 の内部ではクロックスキューが小さく抑えられており、ブロック 48 に含まれる他のフリップフロップとブロック 51 に含まれる他のフリップフロップとの間のパス遅延は、フリップフロップ 46 とフリップフロップ 49 間の信号遅延よりも大きい。フリップフロップ 49 でクロックスキューによるタイミングエラーが起こらないならば、ブロック 51 に含まれる全ての他のフリップフロップで同様にエラーが起こらないと考えてよい。

【0063】ランダムロジック 44 は、クロックスキュー検出回路 53 の出力であるクロックスキュー検出信号 83 を受けて選択回路 3 や出力バッファ回路 4 の制御信号を出力する。つぎに、ランダムロジック 44 およびクロック信号線 5 の駆動回路について説明する。なお、図 4 ではクロック信号線 5 の駆動回路として、遅延回路 2A および出力バッファ回路 4 を設けた構成としているが、第 1 の実施の形態でも説明したように、遅延回路 2A および出力バッファ回路 4 の両方を設けた場合には制御および構成が複雑になるため、以下では、遅延回路 2A および出力バッファ回路 4 のうち一方ずつを設けた 2 つの場合について説明する。

【0064】第 1 の場合として、図 4 の構成において遅延回路 2A を設けていない場合、すなわちクロック信号線 5 の駆動回路として出力バッファ回路 4 のみを設けた場合について説明する。この場合のランダムロジック 44 およびクロック信号線 5 の駆動回路の一例を図 6 に示す。クロック原信号生成回路 1 から出力されたクロック原信号 91 が、バッファ回路 94 および 3 ステートバッファ 95 からなる出力バッファ回路 4A に入力されている。出力バッファ回路 4A の出力線であるクロック信号

線 5 が回路ブロック 51 に接続されている。したがって、回路ブロック 51 のクロック 50 (図 5) はクロック信号線 5 から供給されるクロック信号である。また、クロック 93 および回路ブロック 48 のクロック 47

(図 5) は図示しない他のクロック信号線から供給されるクロック信号であり、クロック 93 には遅延調整は必要なく、クロック 47 には遅延調整が存在する場合も存在しない場合もある。

【0065】ランダムロジック 44 は、入力するクロック 93 を反転させるインバータ回路 98 と、AND (論理積) 回路 99 と、シフトレジスタを構成する複数のフリップフロップ 97 で構成されている。シフトレジスタを構成する先頭のフリップフロップ 97 には論理値

「1」を入力する。また、クロックスキュー検出信号 83 とクロック 93 の反転信号との論理積をフリップフロップ 97 のクロックとして入力する。フリップフロップ 97 のリセット入力にはリセット信号 92 を入力する。フリップフロップ 97 のそれぞれの出力は対応する 3 ステートバッファ 95 の制御入力に接続している。

【0066】なお、この図 6 の出力バッファ回路 4A では、3 ステートバッファではないバッファ回路 94 が存在することが、図 4 に示した出力バッファ回路 4 とは異なるが、バッファ回路 94 を 3 ステートバッファに置き換え、ランダムロジック 44 から常にオン状態に制御する (例えば、先頭のフリップフロップ 97 に入力される論理値「1」を、置き換えた 3 ステートバッファの制御入力に接続する) ことにより、図 4 の出力バッファ回路 4 と同様の構成となる。

【0067】さて、LSI のリセット直後には、全てのフリップフロップ 97 はリセットされて論理値「0」を保持している。その状態では 3 ステートバッファ 95 は全てオフしており、ブロック 51 に供給されるクロック信号線 5 はバッファ回路 94 のみによって駆動される。この時、もしもクロックスキューによるタイミングエラーが起こっていないならば、クロックスキュー検出信号 83 は論理値「0」になり、フリップフロップ 97 にはクロックが入力されない。

【0068】逆にタイミングエラーが起こっている場合には、クロックスキュー検出信号 83 は論理値「1」となり、フリップフロップ 97 にクロックが入力される。その結果、先頭のフリップフロップ 97 は論理値「1」を保持するようになり、対応する 3 ステートバッファ 95 がオンする。その結果、クロック信号線 5 はより高速に駆動されるようになり、クロックスキューは改善される。これを繰り返してクロックスキューが十分小さくなり、タイミングエラーを起こさなくなった時点で、フリップフロップ 97 のシフト動作がストップし、クロックバッファの駆動能力が固定される。

【0069】つぎに第 2 の場合として、図 4 の構成において出力バッファ回路 4 を設けていない場合、すなわち

クロック信号線 5 の駆動回路として遅延回路 2 A のみを設けた場合について説明する。この場合のランダムロジック 4 4 およびクロック信号線 5 の駆動回路の一例を図 7 に示す。クロック原信号生成回路 1 から出力されたクロック原信号 9 1 が遅延回路 2 A に入力され、遅延回路 2 A の出力線であるクロック信号線 5 が回路ブロック 5 1 に接続されている。遅延回路 2 A は、複数のインバータ回路 (1 2 0, 1 2 1 を含む) からなる遅延手段 2 と、遅延手段 2 の出力とランダムロジック 4 4 の出力とをそれぞれ入力する AND (論理積) 回路 1 2 2, 1 2 3, ……、1 2 4 および OR (論理和) 回路 1 2 5 からなる選択回路 3 とで構成されている。

【0070】また、この図 7 のランダムロジック 4 4 は、図 6 のランダムロジック 4 4 の構成にインバータ回路 1 2 6, 1 2 7 および AND 回路 1 2 8 (図示しない省略したものも含む) を付加した構成となっている。この図 7 の場合、LSI のリセット直後には、全てのフリップフロップ 9 7 はリセットされて論理値「0」に設定される。この状態では選択回路 3 の AND 回路 1 2 2 に入力されるインバータ回路 1 2 6 の出力が論理値「1」となり、遅延手段 2 の出力のうち最大に遅延されたインバータ回路 1 2 1 の出力が AND 回路 1 2 2 により選択され、OR 回路 1 2 5 を通ってクロック信号線 5 に出力される。この時、もしもクロックスキューによるタイミングエラーが起こっていないならば、クロックスキュー検出信号 8 3 は論理値「0」になり、フリップフロップ 9 7 にはクロックが入力されない。

【0071】逆にタイミングエラーが起こっている場合には、クロックスキュー検出信号 8 3 は論理値「1」となり、フリップフロップ 9 7 にクロックが入力される。その結果、先頭のフリップフロップ 9 7 に論理値「1」が設定され、今度は選択回路 3 の AND 回路 1 2 3 に入力される AND 回路 1 2 8 の出力が論理値「1」となり、インバータ回路 1 2 0 の出力が AND 回路 1 2 3 により選択され、OR 回路 1 2 5 を通ってクロック信号線 5 に出力される。このときのクロック信号線 5 のクロックは、前述のインバータ回路 1 2 1 の出力が選択されたときのクロックよりも高速であるので、クロックスキューが改善される。

【0072】以上の動作を繰り返して順番により高速なクロックを選択することにより、クロックスキューを十分小さくしてタイミングエラーを回避することができる。なお、タイミングエラーを起こさなくなった時点で、フリップフロップ 9 7 のシフト動作がストップする。以上のように本実施の形態によれば、クロックスキュー検出回路 5 3 により、回路ブロック 4 8, 5 1 内に設けたテスト用のフリップフロップ 4 6, 4 9 に入力されるクロック信号 4 7, 5 0 間のスキューを検出し、その検出したクロック信号間のスキューに応じてクロック信号線 5 の駆動回路を制御するようにしたことにより、

クロック信号に係わるタイミングエラーを回避することができる。従って、従来のように回路修正/再試作を行う必要がなくなり、開発期間/工数を大幅に削減することが可能である。また、製造時の素子特性のばらつきやチップ使用時の電源電圧変動による遅延特性の変化を補償することができるので、チップ歩留まりを改善することも可能である。

【0073】なお、本実施の形態では、フリップフロップ 4 6 とフリップフロップ 4 9 間はゲートを介さずに配線で直結しているが、これは必ずしもそうである必要はない。また、データ発生回路 4 5 やデータ解析回路 5 2、ランダムロジック 4 4 は本実施の形態で例示したものの以外にも様々な回路を考えることができる。

〔第 3 の実施の形態〕次に、本発明の第 3 の実施の形態について以下に説明する。

【0074】図 8 は本発明の第 3 の実施の形態の半導体集積回路の構成図である。図 8 において、クロック原信号生成回路 1、遅延手段 2、選択回路 3、出力バッファ回路 4、クロック信号線 5 については、第 1 の実施の形態と基本的に同じであるため、説明を省略する。本実施の形態においては、第 1 の実施の形態や第 2 の実施の形態とは異なり、クロック信号線 5 の駆動回路 (遅延回路 2 A, 出力バッファ回路 4) は、ランダムロジック 2 3 を介して遅延評価回路 3 4 が出力するデータにより制御される。

【0075】以下、遅延評価回路 3 4 について詳しく説明する。遅延評価回路 3 4 は、データ発生回路 2 4、フリップフロップ 2 5、フリップフロップ 3 0、インバータ素子 2 6、抵抗素子 2 7、容量素子 2 8 およびデータ解析回路 3 3 から構成される。遅延評価回路 3 4 の例を図 9 に示す。データ発生回路 2 4 としては、例えば図 9 に示すようにフリップフロップ 2 5 の出力をインバータ素子 1 0 1 で反転させて入力にフィードバックさせた回路を用いることができる。データ発生回路 2 4 から出力された信号は、クロック 3 5 に同期して、フリップフロップ 2 5 にラッチされる。各フリップフロップ 2 5 からの信号は、インバータ素子 2 6、抵抗素子 2 7、容量素子 2 8 を順番に並べたそれぞれのインバータチェーン 1 0 2, 1 0 5, 1 0 8 を通過する内に、インバータ素子 2 6 の駆動能力、抵抗素子 2 7 の抵抗値、容量素子 2 8 の容量値によって決まるある時間だけ遅延して、それぞれのフリップフロップ 3 0 に到達する。それぞれのインバータチェーン 1 0 2, 1 0 5, 1 0 8 とフリップフロップ 2 5 とフリップフロップ 3 0 とで 1 本の遅延パスを構成する。各フリップフロップ 3 0 もクロック 3 5 に同期して信号をラッチするので、もしも信号が各インバータチェーン 1 0 2, 1 0 5, 1 0 8 を通過して各フリップフロップ 3 0 まで到達するのに要する遅延時間がクロック 3 5 のサイクル時間よりも小さければ、フリップフロップ 3 0 は 1 サイクル前にフリップフロップ 2 5 にラ

ッチされたデータをラッチすることができる。逆に、遅延時間がクロック 35 のサイクル時間よりも大きければ、フリップフロップ 30 は 1 サイクル前にフリップフロップ 25 にラッチされたデータをラッチすることができない。データ解析回路 33 によりデータが正しくフリップフロップ 30 にラッチできたかどうかを判定する。本実施の形態では、データ解析回路 33 としては、例えば図 9 に示すように、各フリップフロップ 25 と各フリップフロップ 30 の出力を入力とする EXNOR 素子 103、106、109 を用いて構成している。

【0076】遅延評価回路 34 には、前述の遅延パスが複数本設けてあり、各々の遅延パスのインバータチェーン 102、105、108 はインバータの段数がそれぞれ異なって構成されている。そのため各々の遅延パスでフリップフロップ 25、30 間の遅延時間が異なるために、複数の遅延パスはデータが正しくラッチできるものと失敗するものとに分かれるが、その境界の遅延パスの遅延値がクロック 35 のサイクル時間におおむね一致する。

【0077】各遅延パスにはチップ設計時に SPICE シミュレーションにより求めた遅延値が分かっており、両者の違いが設計時に想定した遅延と現在のチップ上の遅延の差を表している。これらの違いの原因としては、設計時に用いた回路パラメータと現実のデバイスとの違いや、製造工程上のばらつき、あるいは、チップの温度や電源電圧の違いといったことが考えられる。しかし、いずれの要因もチップ上では均一に作用すると考えられるので、遅延評価回路 34 の遅延値が設計値からはずれているということは、チップ上の他の回路の遅延値も同様にずれていると考えてよい。すなわち、遅延評価回路 34 の遅延値を標準遅延値として、その設計値からのずれに基づいてクロック信号線 5 の駆動回路を制御することで、クロックに係わるタイミングエラーを自動的に回避することが可能である。なお、標準遅延値は、遅延評価回路 34 のインバータチェーンの遅延値のことであり、チップ上の他の回路の遅延値と連動しており、チップ全体の動作時の遅延値を代表するものであり、チップの動作状況が変われば標準遅延値も変化するが、それとは独立してクロック 35 のサイクルは一定である。

【0078】つぎに、遅延評価回路 34 によるクロック信号線 5 の駆動回路の制御について説明する。なお、図 8 ではクロック信号線 5 の駆動回路として、遅延回路 2A および出力バッファ回路 4 を設けた構成としているが、第 1、第 2 の実施の形態でも説明したように、遅延回路 2A および出力バッファ回路 4 の両方を設けた場合には制御および構成が複雑になるため、以下では、遅延回路 2A および出力バッファ回路 4 のうち一方ずつを設けた 2 つの場合について説明する。

【0079】第 1 の場合として、図 8 の構成において遅延回路 2A を設けていない場合、すなわちクロック信号

線 5 の駆動回路として出力バッファ回路 4 のみを設けた場合について説明する。この場合の遅延評価回路 34、ランダムロジック 23 およびクロック信号線 5 の駆動回路の一例を図 9 に示す。この例では、遅延評価回路 34 のそのままの出力により出力バッファ回路 4 を制御しており、EXNOR 素子 103、106、109 の出力で直接 3 ステートバッファ 4a、4b、4c を制御している。クロック原信号生成回路 1 から出力されたクロック原信号 91 が、出力バッファ回路 4 に入力されている。出力バッファ回路 4 の出力であるクロック信号線 5 は図示しない回路ブロックに接続されている。なお、クロック 35 はクロック原信号生成回路 1 から供給されている。

【0080】LSI 設計時には、インバータチェーン 105 を含む遅延パスがクロック 35 のサイクル時間に一致するようにインバータの段数を調整しておく。インバータチェーン 102 はインバータチェーン 105 よりもインバータが 2 段多く、そのため遅延時間はクロック 35 のサイクル時間を越えてしまう。インバータチェーン 108 は逆にインバータチェーン 105 よりもインバータが 2 段少ない。このような状況では、EXNOR 素子 103 は論理値「1」を出力し、EXNOR 素子 106 と EXNOR 素子 109 は論理値「0」を出力する。従って、3 ステートバッファ 4a のみがオンとなり、3 ステートバッファ 4b と 3 ステートバッファ 4c はオフになっている。結局、1 個のバッファでクロック信号線 5 を駆動することになる。

【0081】ところが、この LSI が製造された後、実際に使用された状況では、チップの遅延値が設計時に想定していた遅延値よりも少し大きくなっていくとすると、インバータチェーン 105 を含む遅延パスの遅延もクロック 35 のサイクル時間よりも大きくなることになる。この時には EXNOR 素子 103 と EXNOR 素子 106 が論理値「1」を出力し、EXNOR 素子 109 は論理値「0」を出力する。従って、3 ステートバッファ 4a と 3 ステートバッファ 4b がオンとなり、3 ステートバッファ 4c はオフになる。そのため、2 個のバッファでクロック信号線 5 を駆動することになり、バッファが 1 個の場合よりも高速にクロック信号線 5 を駆動することができる。結局、チップ全体の遅延値の増大に伴ってクロック遅延も増大しようとするのを、クロックバッファを強化することにより補償できたことになる。

【0082】つぎに第 2 の場合として、図 8 の構成において出力バッファ回路 4 を設けていない場合、すなわちクロック信号線 5 の駆動回路として遅延回路 2A のみを設けた場合について説明する。この場合の遅延評価回路 34 (図 9 と同じ構成)、ランダムロジック 23 およびクロック信号線 5 の駆動回路の一例を図 10 に示す。この例では、クロック原信号生成回路 1 から出力されたクロック原信号 91 が遅延回路 2A に入力され、遅延回路

2Aの出力線であるクロック信号線5が図示しない回路ブロックに接続されている。遅延回路2Aは、複数のインバータ回路(130, 131, 132を含む)からなる遅延手段2と、遅延手段2の出力とランダムロジック23の出力とをそれぞれ入力するAND(論理積)回路138~141およびOR(論理和)回路142からなる選択回路3とで構成されている。

【0083】また、この図10のランダムロジック23は、インバータ回路133, 135, 137およびAND回路134, 136から構成されている。この図10の場合、LSI設計時に、前述した図9の場合と同様、遅延評価回路34のインバータチェーン102, 105, 108のインバータの段数を調整しておくことにより、EXNOR素子103は論理値「1」を出力し、EXNOR素子106とEXNOR素子109は論理値「0」を出力する。このとき、AND回路139に入力されるAND回路134の出力が論理値「1」となり、インバータ回路131の出力がAND回路139により選択され、OR回路142を通過してクロック信号線5に出力される。

【0084】ところが、このLSIが製造された後、実際に使用された状況では、チップの遅延値が設計時に想定していた遅延値よりも少し大きくなっていくとすると、インバータチェーン105を含む遅延パスの遅延もクロック35のサイクル時間よりも大きくなることになる。この時にはEXNOR素子103とEXNOR素子106が論理値「1」を出力し、EXNOR素子109は論理値「0」を出力する。このとき、AND回路140に入力されるAND回路136の出力が論理値「1」となり、インバータ回路130の出力がAND回路140により選択され、OR回路142を通過してクロック信号線5に出力される。このときのクロック信号線5のクロックは、前述のインバータ回路131の出力が選択されたときのクロックよりも遅延が小さくなっているため、クロック信号線5の遅延の増大を補償することができる。また逆に、チップの遅延値が設計時に想定していた遅延値よりも小さくなっていった場合には、AND回路138に入力されるインバータ回路133の出力が論理値「1」となり、インバータ回路132の出力がAND回路138により選択され、OR回路142を通過してクロック信号線5に出力されるため、クロック信号線5の遅延の減少を補償することができ、タイミングエラーを回避することができる。

【0085】以上のように本実施の形態によれば、遅延評価回路34により、プロセス変動や使用時における電源電圧変動や温度変化を反映した標準遅延値を検出し、その検出した標準遅延値に応じてクロック信号線5の駆動回路(遅延回路2A, 出力バッファ回路4)を制御するようにしたことにより、クロック信号に係わるタイミングエラーを回避することができる。従って、従来のよ

うに回路修正/再試作を行う必要がなくなり、開発期間/工数を大幅に削減することが可能である。また、製造時の素子特性のばらつきやチップ使用時の電源電圧変動による遅延特性の変化を補償することができるので、チップ歩留まりを改善することも可能である。

【0086】なお、本発明のクロック遅延を調整可能という特徴をより積極的に利用するならば、クロック信号線の駆動回路をハードマクロ化して複数のLSIで共通に使用することも可能である。この場合にはクロック信号線の駆動回路にあらかじめ遅延制御能力を実装しておく必要があるが、従来のようにチップ設計時にクロック信号線の駆動回路をカスタマイズする必要がなくなるため、設計期間/工数を削減することができる。

【0087】また、設計ミスがない場合でも、半導体製造上のばらつきは常に存在しており、チップ歩留まりを低下させる要因となっている。素子特性が標準値からはずれることにより、遅延値が想定した設計マージンを越えて回路が誤動作するような状況に対して、従来のような固定したクロック駆動方式だと対応しようがないため不良品として捨てる以外にない。しかし本発明のようにクロック信号線の駆動回路を可変にしておくことにより、対応できる場合もある。これはLSIの製造歩留まりを改善することにつながる。

【0088】同様にして、上記実施の形態では、あらかじめ想定していなかった電源電圧で使用されるような状況においても、クロック駆動を変更することで対応することが可能である。

【0089】

【発明の効果】以上で説明したように、従来はクロック信号に係わるタイミングエラーが起こるような状況でも、本発明の半導体集積回路ではクロック信号線の駆動回路を自動調整することによりタイミングエラーを回避することができる。従って、従来のように回路修正/再試作を行う必要がなくなり、開発期間/工数を大幅に削減することが可能である。

【0090】また、製造時の素子特性のばらつきやチップ使用時の電源電圧変動による遅延特性の変化を補償することができるので、チップ歩留まりを改善することも可能である。

【図面の簡単な説明】

【図1】本発明の第1の実施の形態の半導体集積回路の構成図。

【図2】本発明の第1の実施の形態の半導体集積回路におけるクロック調整処理のフロー図。

【図3】本発明の第1の実施の形態の半導体集積回路におけるタイミングテストおよび駆動回路制御のフロー図。

【図4】本発明の第2の実施の形態の半導体集積回路の構成図。

【図5】本発明の第2の実施の形態の半導体集積回路に

10 バス信号線

1.1 タイミングテストプログラム

1 2 駆動回路制御プログラム

13 レジスタ

14 クロック

15 レジスタ

16 クロック

- ## 17 被テスト回路

- ## 23 ランダムロジック

- ## 24 データ発生回路

- ## 25 フリップフロップ

- ### 30 フリップフロップ

- ### 3.3 データ解析回路

- ### 3 4 遲延評估回路

- ### 35 クロック

- 44 ランダムロジック

- ## 45 データ発生回路

- #### 4.6 テスト用のフリップフロップ

- 47 クロック

- 20 48 回路ブロック

- #### 49 テスト用のフリップフロップ

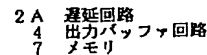
- 50 クロック

- ## 5.1 回路ブロック

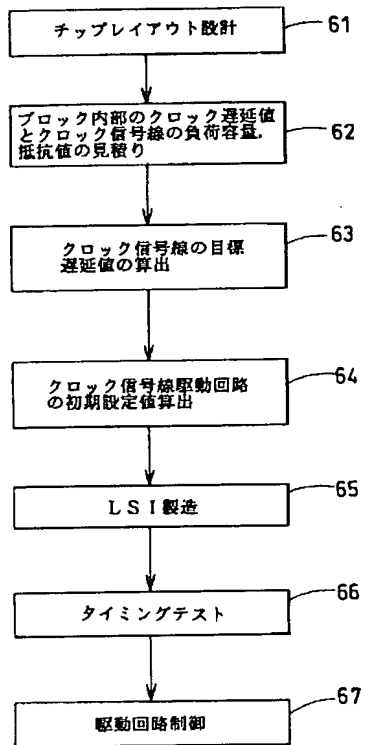
- ## 5.2 データ解析回路

- ### 5.3 クロックスキュー検出回路

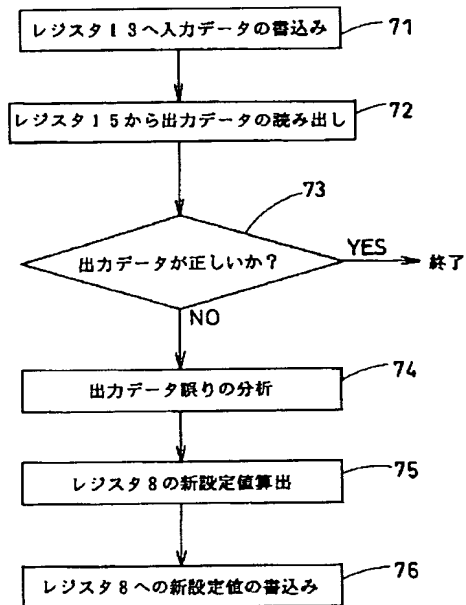
【図 1】



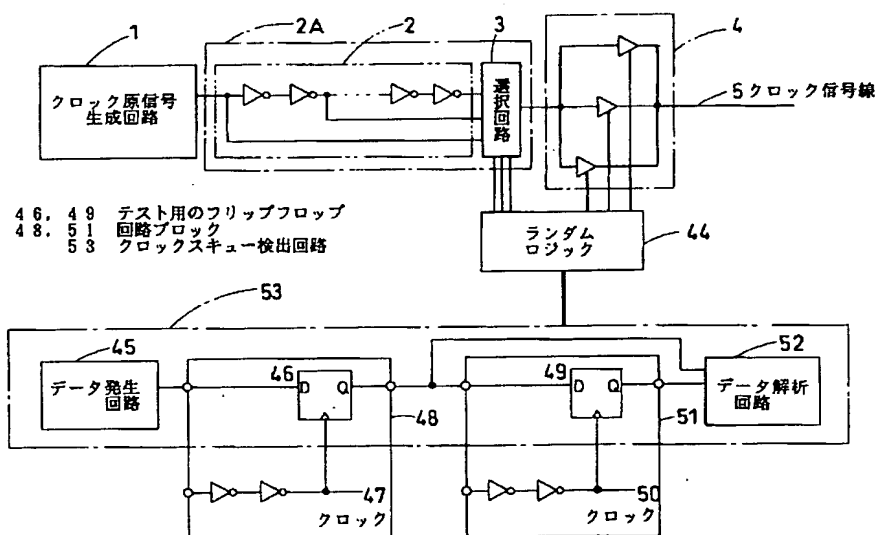
【図 2】



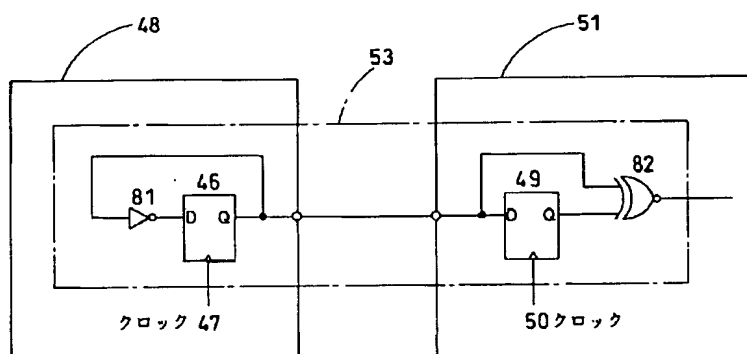
【図 3】



【図 4】



【図 5】



【図 6】

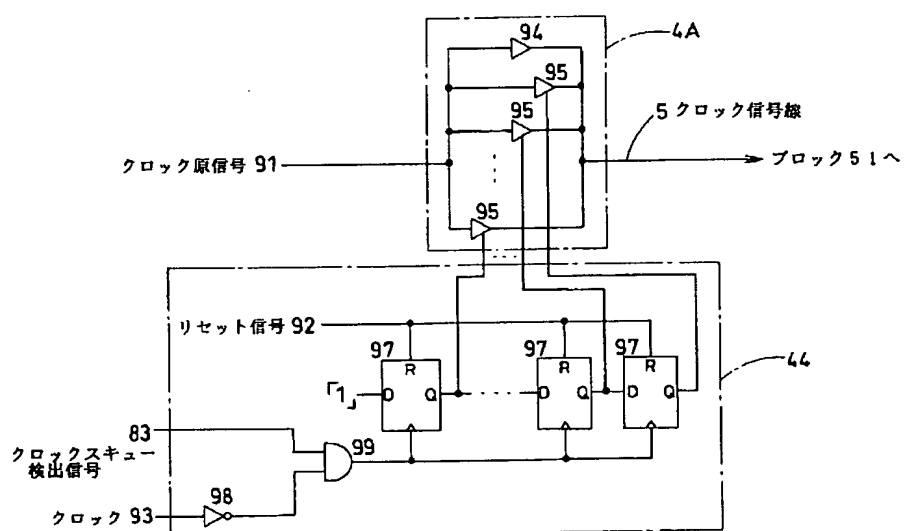


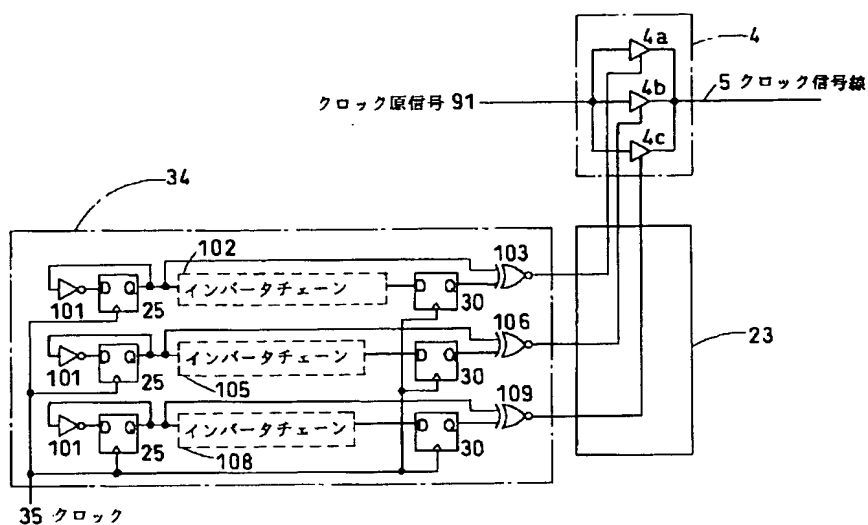
Figure 1 is a block diagram of a clock signal processing circuit. The circuit is divided into two main sections: 2A (top) and 4A (bottom).

Section 2A: This section contains a chain of inverters (120, 121) and a series of AND gates (122, 123, 124) and an OR gate (125). The input to the first inverter (120) is the clock signal 91. The output of the first inverter (120) is connected to the input of the second inverter (121). The output of the second inverter (121) is connected to the input of the first AND gate (122). The output of the first AND gate (122) is connected to the input of the second AND gate (123). The output of the second AND gate (123) is connected to the input of the third AND gate (124). The output of the third AND gate (124) is connected to the input of the OR gate (125). The output of the OR gate (125) is connected to the input of the first AND gate (122).

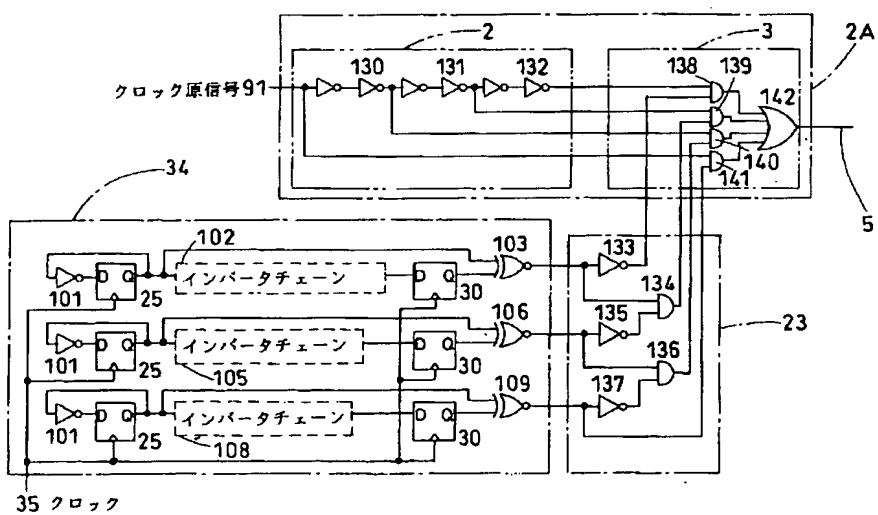
Section 4A: This section contains a chain of D flip-flops (97) and a series of AND gates (99) and OR gates (126, 127, 128). The input to the first D flip-flop (97) is the reset signal 92. The output of the first D flip-flop (97) is connected to the input of the second D flip-flop (97). The output of the second D flip-flop (97) is connected to the input of the third D flip-flop (97). The output of the third D flip-flop (97) is connected to the input of the first AND gate (99). The output of the first AND gate (99) is connected to the input of the second AND gate (99). The output of the second AND gate (99) is connected to the input of the third AND gate (99). The output of the third AND gate (99) is connected to the input of the first OR gate (126). The output of the first OR gate (126) is connected to the input of the second OR gate (127). The output of the second OR gate (127) is connected to the input of the third OR gate (128). The output of the third OR gate (128) is connected to the input of the first D flip-flop (97).

The circuit is controlled by a clock signal (93) and a reset signal (92). The clock signal (93) is connected to the input of the first inverter (120) and the input of the first D flip-flop (97). The reset signal (92) is connected to the input of the first D flip-flop (97).

【図 9】



【図 10】



フロントページの続き

(51) Int. Cl.⁷

H03K 5/13

識別記号

F I

テーマコード* (参考)

F ターム(参考) 5B079 CC02 CC08 CC14 DD06 DD08
DD13 DD20
5F038 BH19 CD06 CD07 CD08 CD09
DF01 DF04 DT02 DT10 DT12
DT17 EZ10 EZ20
5J001 BB00 BB02 BB05 BB08 BB09
BB12 BB13 BB20 BB23 DD01
DD03 DD09